

**DATA BASE ACCESS SYSTEM BY CPU**

Patent Number: JP9179882  
Publication date: 1997-07-11  
Inventor(s): FUKUMOTO MITSUYA  
Applicant(s):: FUJITSU LTD  
Requested Patent: JP9179882  
Application Number: JP19950341568 19951227  
Priority Number(s):  
IPC Classification: G06F17/30 ; G06F12/00  
EC Classification:  
Equivalents:

**Abstract**

**PROBLEM TO BE SOLVED:** To provide the data base access system which enables the CPU to write and read a large amount of data in and out of a data base.

**SOLUTION:** To access respective divisional data bases 1D-nD of matrix constitution of 1st-(n)th data in the format that time information is added to data by 1st-(m)th kinds, respective data storage destination definition tables 37, a table 44 wherein maximum data lengths of kind data are defined for the data numbers by the kinds of the respective data, a table 45 wherein column addresses are defined for the numbers are provided. Here, a 1st converting means 40 recognizes a received data storage destination from the respective data storage destination tables 37, a 3rd converting means 41 recognizes the (Tn)th row address corresponding to the difference time Tn between the head of recognition 1D-nD and reception time information, and a 2nd converting means 42 finds data correspondence numbers by reception kinds from the table 44, finds its number correspondence row address from the table 45, and stores received data in the storage address of the intersection of the column address and the recognition row address of the 3rd converting means 41.

Data supplied from the esp@cenet database - I2

**THIS PAGE BLANK (USPTO)**

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-179882

(43) 公開日 平成9年(1997)7月11日

(51) Int.Cl. <sup>°</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 17/30			G 0 6 F 15/40	3 5 0 C
12/00	5 0 5		12/00	5 0 5
			15/40	3 4 0

審査請求 未請求 請求項の数 6 O L (全 17 頁)

(21) 出願番号 特願平7-341568

(22) 出願日 平成7年(1995)12月27日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番  
1号

(72) 発明者 福元 綱也

神奈川県横浜市港北区新横浜3丁目9番18  
号 富士通コミュニケーション・システム  
ズ株式会社内

(74) 代理人 弁理士 松本 昂

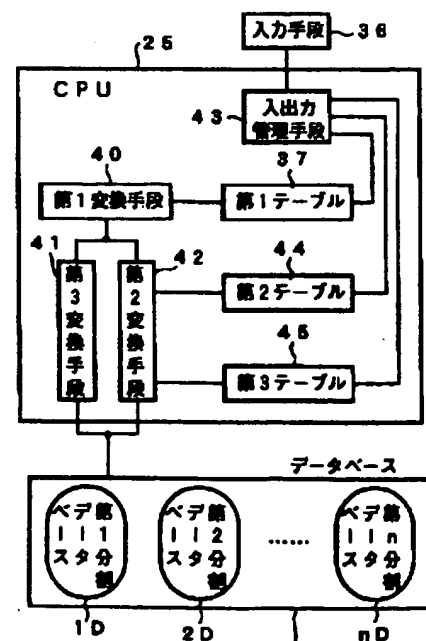
(54) 【発明の名称】 CPUによるデータベースアクセス方式

(57) 【要約】

【課題】本発明はCPUがデータベースに多量のデータを容易に書き込み／読み出すことができるCPUによるデータベースアクセス方式を提供することを目的とする。

【解決手段】第1～第m種別データに時刻情報が付されたフォーマットの第1～第nデータを行列構成の各分割データベース1D～nDにアクセスする際に、各データ記憶先定義テーブル37と、各データの各種別データ番号に種別データ最大データ長を定義したテーブル44と、列アドレスを各番号に定義したテーブル45とを設け、手段40が37から受信データ記憶先を認識し、手段41がその認識1D～nDの先頭と受信時刻情報との差分時間Tnに対応するTn番目の行アドレスを認識し、手段42が44から受信種別データ対応番号を求め、この番号対応列アドレスを45から求め、この列アドレスと手段41での認識行アドレスとの交点の記憶アドレスに受信データを記憶するように構成する。

本発明の原理図



## 【特許請求の範囲】

【請求項1】 各々が、連続した第1～第m種別データに時刻情報が付されたフォーマットの第1～第nデータを一定時間間隔で受信し、CPUの制御によってデータベースを分割した行列構成の第1～第n分割データベースにアクセスするCPUによるデータベースアクセス方式において、

前記第1～第nデータの各々を、前記第1～第n分割データベースの何れに記憶するかを対応付けた第1テーブルと、

該第1～第nデータ中の前記第1～第m種別データに第1～第m種別データ番号を対応付け、該第1～第m種別データ番号に該第1～第m種別データの最大データ長を対応付けた第2テーブルと、

前記第1～第n分割データベースの列方向に順次配置される前記時刻情報の行毎に、順次配列される該第1～第m種別データの各々の配置位置である列アドレスを、該第1～第m種別データ番号に対応付けた第3テーブルと、

前記第1～第nデータの何れかが受信された際に、前記第1テーブルの対応関係から受信データをどの該第1～第n分割データベースに記憶するかを認識する第1変換手段と、

該第1変換手段で認識された該分割データベースの先頭の前記時刻情報から該受信データの時刻情報を減算して差分時間 $T_n$ を求め、該差分時間 $T_n$ に対応する該分割データベースの $T_n$ 番目の行アドレスの行に該受信データを記憶することを認識する第2変換手段と、

該受信データの第1～第m種別データに対応する前記第1～第m種別データ番号を前記第2テーブルから求め、この求められた第1～第m種別データ番号に対応する前記列アドレスを前記第3テーブルから求め、この求められた列アドレスと該第2変換手段で認識された行アドレスとの行列の交点を記憶アドレスとし、この記憶アドレスに該受信データの第1～第m種別データを記憶する第3変換手段とを前記CPUに具備したことを特徴とするCPUによるデータベースアクセス方式。

【請求項2】 入力手段から入力された前記第1～第nデータの名称及び数に応じて、前記第1テーブルに、該第1～第nデータの各々を前記第1～第n分割データベースの何れに記憶するかを対応付ける第1制御を行う入出力管理手段を具備したことを特徴とする請求項1記載のCPUによるデータベースアクセス方式。

【請求項3】 前記入出力管理手段が、前記入力手段から入力された前記第1～第m種別データを前記第2テーブルに配列し、この配列された第1～第m種別データに、前記第1～第m種別データ番号を対応付け、該第1～第m種別データ番号に該第1～第m種別データの最大データ長を対応付ける第2制御を行うことを特徴とする請求項2記載のCPUによるデータベースアクセス方

式。

【請求項4】 前記入出力管理手段が、前記第2制御を行った後、前記第1～第m種別データ番号を前記第3テーブルに配列し、この配列された第1～第m種別データ番号の先頭である第1種別データ番号に前記列アドレスの“0”を対応付け、第2種別データ番号に該“0”と前記第2テーブルの第1種別データの最大データ長との加算値を該列アドレスとして対応付け、以降、対応付けが行われる種別データ番号よりも1つ小さい番号の種別データ番号に対応付けられた該列アドレスの値と該第2テーブルにおける該1つ小さい番号の種別データの最大データ長との加算値を該列アドレスとして対応付ける制御を行うことを特徴とする請求項3記載のCPUによるデータベースアクセス方式。

【請求項5】 前記第1～第nデータにあって前記第1～第m種別データの最大データ長が等しい場合、前記入出力管理手段が、該最大データ長の等しい第1～第nデータに係わる前記第3テーブルを1つのみ定義し、この定義された第3テーブルを前記第3変換手段が共通に用いるようにしたことを特徴とする請求項4記載のCPUによるデータベースアクセス方式。

【請求項6】 前記第2テーブルに定義される前記第1～第m種別データの最大データ長を実データ長としたことを特徴とする請求項1～5の何れかに記載のCPUによるデータベースアクセス方式。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明はCPUによるデータベースアクセス方式に関する。このCPUによるデータベースアクセス方式は、例えば複数の支局から送信されてくる多量のデータを中央局で受信し、中央局のCPUの制御によって縦横の表構成（行列構成）のデータベースに受信データを書き込んで格納すると共に、格納データを読み出して所定の演算処理を行うものである。

【0002】このような処理を行うには、受信データが多量であることからオペレータがデータベースを分割した分割データベースに格納するために受信データ数を把握し、必要な分割データベースの数を算出したり等の作業が必要なので手間がかかったり、CPUが受信データを検索してデータベースの何処に格納するかを認識する処理が必要となったりするので、その処理に時間がかかる。そこで、オペレータの手間を省き、アクセス処理時間を短縮することのできる方式が要望されている。

## 【0003】

【従来の技術】多量のデータを処理する従来のCPUによるデータベースアクセス方式を、ダム諸量システムを例にとって説明する。

【0004】ダム諸量システムは、各地に存在するダム毎に設けられた局舎で収集される情報を更に1つの中央局で収集し、この収集情報をデータベースに格納する構

成となっている。

【0005】図13に1局舎におけるダム諸量システムの概略ブロック構成図を、図14に局舎内のCPUによるデータ受信処理を説明するフローチャートを、図15に正時情報演算用データテーブル図を、図16に局舎内のCPUによる正時情報演算用データテーブルへのデータ格納処理を説明するフローチャートを示す。

【0006】図13において、1は入力装置、2はCPU、3はCPUのメモリ、4はファイル、5はコンソール、6はプリンタ、7はディスプレイ等の卓、8はダムのゲートである。

【0007】最初に、雨量、河川水位、貯水位、開度、放流量等のダム諸量関連のデータは秒単位にて入力装置1に入力され、入力装置1から1分毎に局舎内のCPU2に送信される。

【0008】ここで、CPU2によるダム諸量関連のデータの受信処理を図14に示すフローチャートを参照して説明する。図14のステップS1に示すように、CPU2は1分毎に受信した前記データをメモリ3内の図15に示す正時情報演算用データテーブル10に格納する。

【0009】正時情報演算用データテーブル10には、水位(m)11、貯水量(m<sup>3</sup>)12、放流量(m<sup>3</sup>/sec)13、流入量(m<sup>3</sup>/sec)14、ゲート開度(mm)15、調整流量(m<sup>3</sup>/sec)16等のデータ種別毎に、カウンタ17、積算値18、最大値時刻19、最大値20、最小値時刻21、最小値22、平均値23の領域が確保されている。

【0010】CPU2が1分毎にデータを受信して格納すると、CPU2は図16のフローチャートにおけるステップS1に示すように、全ての入力データの種別毎に以下の処理を行う。

【0011】ステップS2に示すように、まずカウンタ17をインクリメントし、ステップS3に示すように、積算値18に入力データを加算する。次に、ステップS4に示すように、ステップS3で加算された入力データが日情報の最初のデータであるかが判断される。この判断結果がNO(N)の場合は、ステップS5に示すように入力データが最大値20より大きいかな否かが判断される。

【0012】この判断結果、入力データが最大値20より大きいYES(Y)の場合は、ステップS6に示すように、データ入力時刻を最大値時刻19に格納し、ステップS7に示すように入力データを最大値20に格納する。

【0013】次に、ステップS8に示すように、入力データが最小値22より小さいYESの場合は、ステップS9に示すようにデータ入力時刻を最小値時刻21に格納すると共に、ステップS10に示すように入力データを最小値22に格納する。

【0014】また、ステップS4において、日情報の最

初の入力データであるYESの場合は、ステップS12に示すようにデータ入力時刻を最大値時刻19及び最小値時刻21に格納すると共に、ステップS13に示すように入力データを最大値20及び最小値22に格納する。

【0015】このようにして図14のステップS1における格納処理が終了すると、ステップS2に示すように現在が正時(毎時00分)であるかが判断される。この判断結果がNOの場合は全体の処理が終了(END)となるが、YESの場合は、ステップS3に示すように、正時情報演算用データテーブル10のデータを基に正時情報の演算が行われる。即ち、CPU2は1分毎に受信して1時間分集計した情報を基に、毎時00分に正時情報の演算を行っている。

【0016】その正時情報の演算処理を図17に示すフローチャートを参照して説明する。図17のステップS1に示すように、全ての入力データの種別毎に以下の処理を行う。

【0017】ステップS2に示すように、データ種別毎に、積算値18をカウンタ値17にて割った値を平均値23として求め、ステップS3に示すように正時情報演算用データテーブル10に格納する。

【0018】次に、ステップS4に示すように全種別の入力データを演算済であるかを判断し、この判断結果がNOの場合は、ステップS2に戻って再び演算処理を行い、YESの場合は演算を終える。

【0019】このように全データ種別の演算を終了すると、図14のステップS4に示すように、その結果(カウンタ17、積算値18、平均値23)をファイル4(図13参照)に格納すると共に、ステップS5に示すように、中央局にその演算結果を送信する。

【0020】次に、図14のステップS6に示すように、現在が午前9時00分であるかが判断される。この判断結果がNOの場合は全体の処理が終了(END)となるが、YESの場合は、ステップS7に示すように、CPU2がファイル4から1日分の正時情報を読み込んで、ステップS8に示すように日情報の演算を行う。

【0021】即ち、CPU2は毎日午前9時00分に1日分の正時情報(カウンタ値17、積算値18、平均値23のみ)をファイル4からメモリ3に読み込み、この情報を基に日情報の演算を行っている。

【0022】この日情報の演算処理を図18に示すフローチャートを参照して説明する。図18のステップS1に示すように、全ての入力データの種別毎に以下の処理を行う。

【0023】ステップS2に示すように、正時情報のカウンタ値17を合計し、日情報のカウンタ値を算出し、次に、ステップS3に示すように、正時情報の積算値18を合計し、日情報の積算値を算出する。

【0024】次に、ステップS4に示すように、日情報

の積算値を日情報のカウンタ値にて割った値を日情報の平均値として求め、またステップS5に示すように、正時情報演算用データテーブル10内の最大値20、最大値時刻19、最小値22、最小値時刻21を日情報の最大値、最大値時刻、最小値、最小値時刻とし、それぞれの値を図示せぬ日情報演算用データテーブルに格納する。

【0025】次に、ステップS6に示すように全種別の入力データを演算済であるかを判断し、この判断結果がNOの場合は、ステップS2に戻って再び演算処理を行い、YESの場合は演算を終える。

【0026】このように全データ種別の演算を終了すると、図14のステップS9に示すように、その結果をファイル4に格納すると共に、ステップS10に示すように、中央局にその演算結果を送信する。

【0027】次に、ステップS11に示すように、現在が1か月の内の所定日の午前9時00分であるかが判断される。この判断結果がNOの場合は全体の処理が終了(END)となるが、YESの場合は、ステップS12に示すように、CPU2がファイル4から1月分の正時情報を読み込んで、ステップS13に示すように月情報の演算を行う。

【0028】即ち、CPU2は毎月1日午前9時00分に1月分の日情報をファイル4からメモリ3に読み込み、この情報を基に月情報の演算を行っている。この月情報の演算処理を図19に示すフローチャートを参照して説明する。

【0029】図19のステップS1に示すように、全ての入力データの種別毎に以下の処理を行う。ステップS2に示すように、日情報のカウンタ値を合計し、月情報のカウンタ値を算出し、次にステップS3に示すように、日情報の積算値を合計し、月情報の積算値を算出する。

【0030】次に、ステップS4に示すように、月情報の積算値を月情報のカウンタ値にて割った値を月情報の平均値として求め、またステップS5に示すように、日情報の最大値の中で最も大きい値を検索し、月情報の最大値及び最大値時刻とすると共に、ステップS6に示すように、日情報の最小値の中で最も小さい値を検索し、月情報の最小値及び最小値時刻とする。

【0031】次に、ステップS7に示すように全種別の入力データを演算済であるかを判断し、この判断結果がNOの場合は、ステップS2に戻って再び演算処理を行い、YESの場合は演算を終える。

【0032】このように全データ種別の演算を終了すると、図14のステップS14に示すように、その結果をファイル4に格納すると共に、ステップS15に示すように、中央局にその演算結果を送信する。

【0033】次に、ステップS16に示すように、現在が1月1日の午前9時00分であるかが判断される。こ

の判断結果がNOの場合は全体の処理が終了(END)となるが、YESの場合は、ステップS17に示すように、CPU2がファイル4から1年分の正時情報を読み込んで、ステップS18に示すように年情報の演算を行う。

【0034】即ち、CPU2は毎年1月1日午前9時00分に1年分の月情報をファイル4からメモリ3に読み込み、1年分の月情報を基に年情報の演算を行っている。この年情報の演算は、図19を参照して説明した月情報の演算と同一なのでその説明を省く。但し、図19において、“日情報”が“月情報”に、“月情報”が“年情報”に代わる。

【0035】年情報の全データ種別の演算を終了すると、図14のステップS19に示すように、その結果をファイル4に格納すると共に、ステップS20に示すように、中央局にその演算結果を送信する。以上で局舎内全体の処理が終了する。

【0036】また、ファイル4のメモリ容量は格納するデータの種別が多くなればなるほど増えるため、通常保存期間、例えば正時情報は1ヶ月間、日情報は1年間、月情報は2年間、年情報は10年間を決めており、保存期間を過ぎると新しいデータが上書きされるようになっている。

【0037】この方法でファイル4内にデータを格納すると、ファイル4内のメモリ容量はほぼ限界に達する。また、正時情報や日情報は日報に、月情報は月報に、年情報は年報に用いられている。

【0038】また、オペレータがコンソール5の操作によって、CPU2の演算結果をプリンタ6からプリントアウト、また卓7に表示させ、これを参照することによってダム設備の制御、例えばその1つであるゲート8の制御が行われている。

【0039】次に、図20に中央局におけるダム諸量システムの概略ブロック構成図を示し、その説明を行う。図20において、25はCPU、26はCPUのメモリ、27はファイル、28はデータベース、29はコンソール、30はプリンタ、31はディスプレイ等の卓である。

【0040】CPU25には、上述した各地に存在する局舎から定期的にダム諸量データが送信されてくる。そのダム諸量データは、データベース28に格納されるようになっている。図21に行列構成のデータベース28の構成の一例を示す。但し、ダム諸量データの正時情報を格納する構成であるとする。

【0041】中央局では各局舎から定期的に受信しているダム諸量データを、正時情報、日情報、月情報、年情報別に図21に示すようなデータベース28内に格納している。

【0042】このようにデータベース28は横方向にかなり長い。また中央局にダム諸量データを定期的に送信

している局舎の数が増えるほど、更に各局舎が中央局に送信しているダム諸量データの水位、貯水量等のデータ種別（以下データ名と呼ぶ）が増えるほど、データベース28は横方向に長くなる。

【0043】しかし通常のデータベース28には横方向のデータ数（データバイト長）に制限がある。ダム諸量データの数がデータベース28の横方向の許容データ数より小さい場合は、図22に示すCPU25の機能の1つであるアプリケーションプログラム（以下アプリと呼ぶ）33が、特に問題無くデータベース28内にデータを書き込んだり、データベース28内のデータを読み込むことが可能である。

【0044】しかし、ダム諸量データの数がデータベース28の横方向の許容データ数より大きい場合は、予めデータベース28を図22に示すように、横方向に必要なだけ分割し（以下分割した個々のデータベースを表と呼ぶ）、この分割された第1表1D～第n表nDに、アプリ33がデータベースアクセスインタフェース34、データベースアクセス手段35を介してデータを書き込んだり、表1D～表nDからデータを読み込んだりするようになる。

【0045】ここで、CPU25のアプリ33が行う局舎からのダム諸量データの受信処理を図23を参照して説明する。システム立ち上げ時に中央局のオペレータは、図22に示すキーボード36の操作によって、1つの表の横方向に許容可能な最大データ数（以降8192バイトとする）及び、局舎数、1時点に1局舎から受信するダム諸量データの総データ数の最大値等から必要な表の数を算出すると共に、各局舎毎また正時情報、日情報、月情報、年情報が格納されているファイルの種別（以下ファイルNoと呼ぶ）毎にどの表に格納するかを決め、図24に示すようなアプリ33内の表名定義テーブル37に定義する。

【0046】この後、図23のステップS1に示すように、各局舎からダム諸量データを受信する。ここでは局舎Aからのデータが受信されたとする。次にステップS2に示すように、アプリ33は、ファイルNo、局舎名（例えばA）、年月日及び時刻（以下時刻情報と呼ぶ）、水位、貯水量等のデータ名を読み込んだ後に、ステップS3に示すように、表名定義テーブル37より局舎Aの該当ファイルNoの情報が格納されている表名（以下表1Dとする）を読み込むと共に、ステップS4に示すように、データベースアクセスインタフェース34を通して表1Dをアクセスし、表1Dから順に該当時刻情報を捜す。

【0047】次に、ステップS5に示すように、表1Dの該当時刻情報の領域内に受信した全てのデータを1つずつ格納する。この格納手順は、まずステップS6に示すように、データ名を1つ読み込み、ステップS7に示すように、読み込んだデータ名の領域を左から順に検索

し、ステップS8に示すように、読み込んだデータ名に該当する領域に該当する受信データを格納する。

【0048】そして、ステップS9に示すように全てのデータ名に対して格納処理を終えたか否かを判断し、この判断結果が終えていないNOの場合は、ステップS6に戻って処理を継続し、終えたYESの場合は処理を終了する。

【0049】なおデータベース28の縦方向のメモリ容量は格納するデータの種類の多くなればなるほど増えるため、通常保存期間、例えば正時情報は3ヶ月間、日情報は3年間、月情報は20年間、年情報は100年間を決めており、保存期間を過ぎると新しいデータが上書きされるようになっている。このようにデータベース28内にデータを格納すると、データベース28内のメモリ容量はほぼ限界に達する。なお正時情報や日情報は日報に、月情報は月報に、年情報は年報に用いられている。

【0050】

【発明が解決しようとする課題】ところで、上述した中央局におけるCPU25によるデータベースアクセス方式においては、次に記述するような問題点があった。

【0051】①上述したように、システム立ち上げ時に中央局のオペレータは1つの表に許容可能な最大データ数及び局舎数、1時点に1局舎から受信するダム諸量データの総データ数の最大値等から必要な表（分割データベース）の数を算出すると共に、各局舎毎またファイルNo毎にどの表に格納するかを決め、図24に示すようなアプリ33内の表名定義テーブル37に定義しているが、この定義を行うためのオペレータの作業には手間がかかる問題があり、また、オペレータはアプリ33をそれぞれのダム諸量システムに適用する際にはアプリ33内の表名定義テーブル37の内容を書き換える必要があり、この場合、アプリ33の内容をも変更しなおさなければならないので、その作業に手間がかかる問題があった。

【0052】②アプリ33は表名定義テーブル37から表名を読み込んだ後、表の上から順に該当時刻情報を捜す処理を行っているが、図21に示すように表の中には通常かなりの数の時刻情報が存在し、特に下の方の時刻情報の場合、検索に時間がかかる問題があった。

【0053】③アプリ33は表の中から該当時刻情報を検索した後、受信した全てのデータについて1つずつ該当時刻情報の領域内からデータ名の領域を左から順に検索する処理を行っているが、特にデータ名が沢山ある場合、時刻情報の領域内からデータ名の領域を検索するのに時間がかかる問題があった。

【0054】本発明は、このような点に鑑みてなされたものであり、CPUがデータベースに多量のデータを容易に書き込み/読み出すことができるCPUによるデータベースアクセス方式を提供することを目的としている。

## 【0055】

【課題を解決するための手段】図1に本発明の原理図を示す。この図に示すCPUによるデータベースアクセス方式は、各々が、連続した第1～第m種別データに時刻情報が付されたフォーマットの第1～第nデータを一定時間間隔で受信し、CPU25の制御によってデータベース28を分割した行列構成の第1～第n分割データベース1D、2D、…、nDにアクセスするものである。

【0056】本発明の特徴は、CPU25に、第1～第nデータの各々を、第1～第n分割データベース1D～nDの何れに記憶するかを対応付けた第1テーブル37と、第1～第nデータ中の第1～第m種別データに第1～第m種別データ番号を対応付け、第1～第m種別データ番号に第1～第m種別データの最大データ長を対応付けた第2テーブル44と、第1～第n分割データベース1D～nDの列方向に順次配置される時刻情報の行毎に、順次配列される第1～第m種別データの各々の配置位置である列アドレスを、第1～第m種別データ番号に対応付けた第3テーブル45と、第1～第nデータの何れかが受信された際に、第1テーブル37の対応関係から受信データをどの第1～第n分割データベース1D～nDに記憶するかを認識する第1変換手段40と、第1変換手段40で認識された分割データベース1D～nDの先頭の時刻情報から受信データの時刻情報を減算して差分時間Tnを求め、差分時間Tnに対応する分割データベース1D～nDのTn番目の行アドレスの行に受信データを記憶することを認識する第2変換手段41と、受信データの第1～第m種別データに対応する第1～第m種別データ番号を第2テーブル44から求め、この求められた第1～第m種別データ番号に対応する列アドレスを第3テーブル45から求め、この求められた列アドレスと第2変換手段41で認識された行アドレスとの行列の交点を記憶アドレスとし、この記憶アドレスに受信データの第1～第m種別データを記憶する第3変換手段42とをCPU25に具備して構成したことにある。

【0057】このような構成によれば、大規模な行列構成のデータベース1D～nDであっても、受信データの記憶アドレスを容易に検索することができるので、従来のような、順列に従った検索を行うことによって時間がかかるといったことを解消することができる。

## 【0058】

【発明の実施の形態】以下、図面を参照して本発明の一実施の形態について説明する。図2は本発明の一実施形態のCPUによるデータベースアクセス方式を適用したダム諸量システムにおける中央局のCPU機能ブロック構成図である。この図2において図22に示した従来例の各部に対応する部分には同一符号を付し、その説明を省略する。

【0059】図2に示す一実施形態が図22に示した従来例と異なる点は、CPU25に新たに、第1変換手段

(表名変換ツール)40、第2変換手段(レコードNo変換ツール)41、第3変換手段(データ名No変換ツール)42、入出力管理プログラム43、第2テーブル(データ名No定義テーブル)44、及び第3テーブル(列名定義テーブル)45を設け、従来アプリ33内にあった第1テーブル(表名定義テーブル)37をアプリ33外に設けたことにある。

【0060】図2に示す実施形態のアプリには、従来例のアプリ33と区別するために符号33'を付す。図3に入出力管理プログラム43がダム諸量システム立ち上げ時に行う処理動作を説明するためのフローチャートを示し、その説明を行う。

【0061】図3のステップS1に示すように、システム立ち上げ時に中央局のオペレータが入力手段(キーボード)36から1つの表(分割データベース)に許容可能な最大データ数及び局舎数、1時点に1局舎から受信するダム諸量データの総データ数の最大値及び受信する可能性のあるダム諸量データの全てのデータ名等の情報を入力すると、それを入出力管理プログラム43が受信する。

【0062】これによって、ステップS2に示すように、入出力管理プログラム43がステップS1の入力値よりデータベース28における必要な表の数を算出すると共に、ステップS3に示すように、各局舎毎またファイルNo毎にどの表に格納するかを決め、図24に示したような表名定義テーブル37に定義する。

【0063】次に、ステップS4に示すように、受信する可能性のあるダム諸量データの全てのデータ名等の情報から各データ名毎にデータ名Noを決め、最大データ長と共に図4にその一例を示すデータ名No定義テーブル44に格納する。

【0064】次に、ステップS5に示すように、全ての局舎に対し、各局舎毎また各ファイルNo毎に下記のステップS6～S10に示す手順にて、図5～図7にその例を示す列名定義テーブル45を定義する。

【0065】但し、図5は図24に示す局舎AのファイルNoが正時情報の場合の列名定義テーブル45の一例図、図6は図24に示す局舎CのファイルNoが正時情報の場合の列名定義テーブル45の一例図、図7は図24に示す局舎FのファイルNoが正時情報の場合の列名定義テーブル45の一例図である。

【0066】ステップS6に示すように、図24に示す表名定義テーブル37より各局舎毎また各ファイルNo毎に定義された表名を左上から順に読み込む。次に、ステップS7に示すように、ステップS6で読み込んだ局舎名及びファイルNoに対して下記のステップS8及びS9に示す手順にて列名定義テーブル45の全てのデータ名Noについて各データNo毎に該当表内の列名を定義する。

【0067】即ち、ステップS8に示すように、該当列名定義テーブル45に最後に定義した列名に最後に定義



したデータNoの最大データ長（データ名No定義テーブル44より読み込む）を加算して該当データ名Noの該当表内の列名を算出し、列名定義テーブル45内の該当データ名Noの列名領域に格納する。但し、最初の定義の場合の列名は自動的に“0”となるようになっている。

【0068】例えば、図4に示すデータ名No定義テーブル44のデータ名Noが“1”の場合、その“1”は最初に図5に示す列名定義テーブル45に列名を定義するためのデータ名Noなので、テーブル45の列名は“0”となる。

【0069】次に、データ名No“2”に定義される列名は、その“0”にデータ名No“1”の最大データ長“180”を加算した“180”となる。その次に、データ名No“3”に定義される列名は、その“180”にデータ名No“2”の最大データ長“175”を加算した“355”となる。以降同様に列名が定義される。

【0070】次に、ステップS9に示すように、全てのデータ名Noについて列名の格納処理を終えたか否かを判断し、この判断結果が終えていないNOの場合は、再びステップS8の処理を行い、終えたYESの場合はステップS10に進む。

【0071】そして、ステップS10に示すように、全ての局舎及び全てのファイルNoについて格納処理を終えたか否かを判断し、この判断結果が終えていないNOの場合は、ステップS6に戻り、終えたYESの場合は処理を終了する。

【0072】このような入出力管理プログラム43の処理動作によれば、システム立ち上げ時に中央局のオペレータが入力手段（キーボード）36から1つの表に許容可能な最大データ数及び局舎数、1時点に1局舎から受信するダム諸量データの総データ数の最大値及び受信する可能性のあるダム諸量データの全てのデータ名等の情報を入力するだけで、次に記述するような効果がある。

【0073】即ち、従来のように、システム立ち上げ時に中央局のオペレータが1つの表に許容可能な最大データ数及び局舎数、1時点に1局舎から受信するダム諸量データの総データ数の最大値等から必要な表の数を算出する必要がなくなると共に、各局舎毎またファイルNo毎にどの表に格納するかを決め、図24に示すようなアプリ33内の表名定義テーブル37に定義する必要がなくなる。

【0074】また、オペレータがアプリ33をそれぞれのダム諸量システムに適用する際にアプリ33内の表名定義テーブル37の内容を書き換える必要がなくなるので、アプリ33の内容をその書換えの都度変更しなおす必要がなくなる。

【0075】次に、上述したシステム立ち上げを終えた後、実際に、ある局舎（以降局舎Aとする）からダム諸量データを受信した場合にアプリ33'が行う処理手順について図8に示すフローチャートを参照して説明す

る。

【0076】まず、ステップS1に示すように、局舎Aからダム諸量データを受信したとする。この受信後にアプリ33'は、ステップS2に示すように、受信データよりファイルNo（以降正時情報とする）、局舎名（局舎A）、時刻情報、データ名を読み込む。

【0077】この後に、ステップS3に示すように、表名変換ツール40を用いて表名定義テーブル37より局舎Aの該当ファイルNo（正時情報）の情報が格納されている表名（以下表1Dとする）を読み込む。

【0078】次に、ステップS4に示すように、データベースアクセスインタフェース34を介して表1Dをアクセスした後、レコードNo変換ツール41を用いて表1Dの中から該當時刻情報のアドレスを算出する。

【0079】例えば受信データの局舎名を局舎A、ファイルNoを正時情報とし、また局舎名が局舎AでファイルNoが正時情報に該当する表の先頭に格納された受信データの時刻情報が図21に示す1月1日10:00であるとする。

【0080】図21に示すように、以降1時間毎に正時情報のデータを受信するので、受信したデータの時刻情報と、表の先頭に格納された受信データの時刻情報との時間差から表の何行目に格納すべきかを算出することが可能である。

【0081】例えば2月1日11:00の時刻情報のデータを受信した場合、表の先頭に格納された受信データの時刻情報との時間差は、31日と1時間、即ち745時間であり、表の746行目に格納すべきであることを算出できる。

【0082】但し、データベース28内のデータには通常保存期間がある。新しく受信したデータはデータベース28に上書きされているが、表の先頭に格納されているデータが丁度、保存期間だけ前の場合は表の先頭に上書きされることになる。

【0083】これによって、従来の表の上から順に該當時刻情報を検索するのに時間がかかる問題を解消することができる。次に、ステップS5に示すように、ステップS4で算出した表1D内の該當時刻情報のアドレスの領域内に受信した全てのデータ名をデータ名No変換ツール42を用いて1つずつ格納する。この格納手順をステップS6～S10にて説明する。

【0084】ステップS6に示すように、受信したデータ名を1つ読み込む。この後、ステップS7に示すように、データ名No変換ツール42を用いて読み込んだデータ名に該当するデータ名Noをデータ名No定義テーブル44より読み込む。

【0085】次に、ステップS8に示すように、データ名No変換ツール42を用いて局舎AのファイルNo（正時情報）に該当する列名定義テーブル45より読み込んだデータ名Noに該当する列名を読み込み、ステップS9に

示すように、表1Dの該当時刻情報のアドレスの領域内の読み込んだ列名に該当する領域に受信データを格納する。

【0086】例えば受信データの局舎名を局舎A、ファイルNoを正時情報とする。受信したデータ名の水位・貯水量・放流量・流入量・ゲート開度・調整流量…を1つずつ読み込み、データ名No変換ツール42を用いて読み込んだデータ名に該当するデータ名Noの1・2・3・4・5・6…を、図4に示すデータ名No定義テーブル44より読み込む。

【0087】更にデータ名No変換ツール42を用いて局舎AのファイルNoが正時情報に該当する図5に示す列名定義テーブル45より読み込んだデータ名Noに該当する列名の0・180・355・525・695・865…を読み込み、読み込んだ列名に該当する図21に示す列の領域（水位・貯水量…）に受信データを格納する。これにより、従来の時刻情報の領域内からデータ名を検索するのに時間がかかる問題を解消することができる。

【0088】最後に、ステップS10に示すように、全てのデータ名について格納処理を終えたか否かを判断し、この判断結果が終えていないNOの場合は、ステップS6に戻り、終えたYESの場合は処理を終了する。

【0089】このようなレコードNo変換ツール41及びデータ名No変換ツール42を用いたアプリ33の格納処理によれば、従来において、アプリ33が図22に示したように自己内に設けられた表名定義テーブル37から表名を読み込んだ後、図21に示したかなりの数の時刻情報が存在する表の上から順に該当時刻情報を検索する際に、特に下の方の時刻情報の場合に検索に時間がかかるといったことを解消することができる。

【0090】更に、従来のアプリ33が先の表の中から該当時刻情報を検索した後、受信した全てのデータについて1つずつ該当時刻情報の領域内からデータ名の領域を左から順に検索する際に、特にデータ名が沢山ある場合、時刻情報の領域内からデータ名の領域を検索するのに時間がかかるといったことを解消することができる。

【0091】次に、定義する列名定義テーブル45の数を削減することによってCPU25のメモリ量の削減を図る場合の説明を行う。システム立ち上げ時に、図8のステップS5～S10において説明したように、入出力管理プログラム43は全ての局舎に対し、各局毎又は各ファイルNo毎に列名定義テーブル45をCPU25のメモリ内に定義しているが、ここで局舎数が増えるほど、またデータ名の種類が増えるほどメモリ量は増えていく。しかし当然のことながらメモリ量には限界があり、メモリ容量オーバーになることもあり得る。

【0092】図5及び図6に示す局舎Aと局舎Cの正時情報のファイルNoの列名定義テーブル45は、表は異なるが中身は全く同じである。図24の表名定義テーブル37を見ると、1つの表には同一ファイルNoの2つの局

舎のデータを格納するようになっている。また図4のデータ名No定義テーブル44を見ると、全ての局舎のデータや全てのファイルNoのデータにおいて、同一データ名の最大データ長は同一に定義されているため、全ての列名定義テーブル45の中身は2つのパターンしか存在しない。即ち、図5及び図6に示す列名定義テーブル45のパターンと、図7に示す列名定義テーブル45のパターンしか存在しないことになる。

【0093】よって上記のケースではシステム立ち上げ時に、入出力管理プログラム43が列名定義テーブル45を2種類のみ定義するようにすればよい。また実際に受信データを表に格納する際には局舎名からどちらのパターンの列名定義テーブル45から列名を読み込めばよいかを判断した後に、データ名No変換ツール42を用いて読み込んだデータ名Noに該当する列名を読み込めばよい。

【0094】これによって、定義する列名定義テーブル45の数を大幅に削減することができるので、CPU25のメモリ量を削減することができる。次に、ファイルNoの種別毎にデータ名No定義テーブル44を定義することによってデータベース28内のメモリ量の削減を図る場合の説明を行う。

【0095】例えば図7に示す列名定義テーブルの場合、同一の局舎で同一のファイルNoの受信データをデータベース28内の表に格納する場合、横方向に3200バイト必要なことが分かる。

【0096】1つの表の横方向に許容可能な最大データ数が8192バイトとすると、図24に示すように、横方向には同一の局舎で同一のファイルNoの受信データを2種類しか格納できないが、ここで局舎数が増えるほど、またデータ名の種類が増えるほどデータベース28内のメモリ量は増えていく。しかし当然のことながらメモリ量には限界があり、メモリ容量オーバーになることもあり得る。

【0097】図4に示すデータ名No定義テーブル44においては、全ての局舎のデータや全てのファイルNoのデータにおいて、同一データ名の最大データ長は同一に定義されている。

【0098】この最大データ長は、各データ名毎に全局舎及び全ファイルNoのデータの中の最大のデータ長であり、局舎名やファイルNoが異なれば各データ名毎に最大データ長は異なる。

【0099】例えばファイルNoについて考えてみると、積算値、カウンタ値については累計されていくために年情報のファイルNoの最大データ長が最も長く、正時情報の最大データ長が最も短くなる。また、局舎名について考えてみると、ダムによって当然水位、貯水量等の値は異なるために最大データ長も異なる。

【0100】図9にデータ名No定義テーブルの他例図、図10に図24に示す局舎AファイルNoが正時情報の場

10

20

30

40

50

合の列名定義テーブル45の他例図、図11に図24に示す局舎BファイルNoが正時情報の場合の列名定義テーブル45の他例図、図12に図24に示す局舎CファイルNoが正時情報の場合の列名定義テーブル45の他例図を示す。

【0101】図9に示すデータ名No定義テーブル44においては、ファイルNoによって同一データ名でも最大データ長を変えた。これによって正時情報のファイルNoについては1つの表の横方向に局舎A、B、Cの3局舎分の領域を確保することができ、これによりデータベース28内のメモリ量の削減につながる。

【0102】但し、図9に示すデータ名No定義テーブル44では全ての局舎のデータにおいて、同一データ名の最大データ長は同一に定義してあるが、局舎によって変えることも可能である。

【0103】なお、このような処理を用いると、前述したように削減する列名定義テーブル45の数を、システム立ち上げ時にCPU25のメモリ上に必要なだけ増やす必要がある。次に、データベース28の横方向に許容可能なデータ数が同一局舎、同一ファイルNoの受信データ数より小さい場合のデータベース28の分割処理について説明する。

【0104】今まではデータベース28の横方向に許容可能なデータ数が同一局舎、同一ファイルNoの受信データ数より大きい場合、即ち、図5～図7では2局舎分の受信データを、図10～図12では3局舎分の受信データをデータベース28内の同一行に格納可能である。

【0105】しかし、同一局舎、同一ファイルNoの受信データ数が極端に多かったり、またデータベース28の横方向に許容可能なデータ数が少ない場合、同一局舎、同一ファイルNoの受信データでもデータベース28内の同一行に格納できない場合があり得る。

【0106】この場合、システム立ち上げ時に入出力管理プログラム43が、中央局のオペレータがキーボード36から入力した、1つの表に許容可能な最大データ数及び局舎数、1時点に1局舎から受信するダム諸量データの総データ数の最大値及び受信する可能性のあるダム諸量データの全てのデータ名等の情報を受け取った後に、これら入力値より必要な表の数を図3に示すステップS2に示すように算出する。

【0107】しかし、この時点で入出力管理プログラム43が、同一局舎、同一ファイルNoの受信データでもデータベース28内の同一行に格納できないと判断し、以降同一局舎及び同一ファイルのデータ名毎にどの表に格納するかを決め、表名定義テーブル37に定義することになるが、この場合の表名定義テーブル37の構成は図24とは異なり、各局舎毎に作成され、縦軸には図24と同様にファイルNoが、また横軸には新たにデータ名がとられる。

【0108】その後、図3のステップS6に示すよう

に、その表名定義テーブルの左上より表名を読み込むが、各局舎毎また各ファイルNo毎また各データ名毎に読み込むことになる。

【0109】これにより、データベース28の横方向に許容可能なデータ数が同一局舎、同一ファイルNoの受信データ数より小さい場合でも表の分割及びデータベース28への格納が可能となる。

【0110】以上説明した実施形態のCPUによるデータベースアクセス方式によれば以下に記述するような効果がある。通常のデータベース28には横方向のデータ数（データバイト長）に制限がある。各局舎が収集し、中央局に送信されたダム諸量等のデータを中央局のCPUがデータベース28内に格納する際に、同一時刻の同一ファイルNoや同一局舎のデータは、通常データベース28内の同一行に格納しているが、データベース28の横方向の許容可能な最大データ数がデータベース28内の同一行に格納したいデータ数よりも小さい場合でも、アプリ33'はデータベースアクセスインタフェース34を特に意識することなくデータベース28をアクセスすることが可能である。

【0111】また、アプリ33'は、受信データをデータベース28内に格納する際に、時刻情報をサーチすることなくそのアドレスを算出することができ、またデータ名をサーチすることなくその列名を読み込むことができる。これによって、CPU25の負荷軽減及び処理時間の短縮が可能となる。

【0112】

【発明の効果】以上説明したように、本発明によれば、CPUがデータベースに多量のデータを容易に書き込み／読み出すことができる効果がある。

【図面の簡単な説明】

【図1】本発明の原理図である。

【図2】本発明の一実施形態のCPUによるデータベースアクセス方式を適用したダム諸量システムにおける中央局のCPU機能ブロック構成図である。

【図3】入出力管理プログラムがシステム立ち上げ時に行う処理動作を説明するためのフローチャートである。

【図4】データ名No定義テーブル例図である。

【図5】図24に示す局舎AのファイルNoが正時情報の場合の列名定義テーブル例図である。

【図6】図24に示す局舎CのファイルNoが正時情報の場合の列名定義テーブル例図である。

【図7】図24に示す局舎FのファイルNoが正時情報の場合の列名定義テーブル例図である。

【図8】局舎Aからのダム諸量データ受信時にCPUのアプリケーションプログラムが行う処理動作を説明するフローチャートである。

【図9】データ名No定義テーブルの他例図である。

【図10】図24に示す局舎AのファイルNoが正時情報の場合の列名定義テーブルの他例図である。

17

【図11】図24に示す局舎BのファイルNoが正時情報の場合の列名定義テーブル例図である。

【図12】図24に示す局舎CのファイルNoが正時情報の場合の列名定義テーブルの他例図である。

【図13】1局舎におけるダム諸量システムの概略ブロック構成図である。

【図14】局舎内のCPUによるデータ受信処理を説明するためのフローチャートである。

【図15】正時情報演算用データテーブル図である。

【図16】局舎内のCPUによる正時情報演算用データテーブルへのデータ格納処理を説明するためのフローチャートである。

【図17】局舎内のCPUによる正時情報演算処理を説明するためのフローチャートである。

【図18】局舎内のCPUによる日時情報演算処理を説明するためのフローチャートである。

【図19】局舎内のCPUによる月時情報演算処理を説明するためのフローチャートである。

【図20】中央局におけるダム諸量システムの概略ブロック構成図である。

18

【図21】中央局のCPUのデータベースの一構成例図である。

【図22】従来の中央局のCPU機能ブロック構成図である。

【図23】従来の中央局のCPUによるダム諸量データの受信処理を説明するためのフローチャートである。

【図24】表名定義テーブル図である。

【符号の説明】

25 CPU

28 データベース

36 入力手段(キーボード)

37 第1テーブル(表名定義テーブル)

40 第1変換手段(表名変換ツール)

41 第2変換手段(レコードNo変換ツール)

42 第3変換手段(データ名No変換ツール)

43 入出力管理手段(入出力管理プログラム)

44 第2テーブル(データ名No定義テーブル)

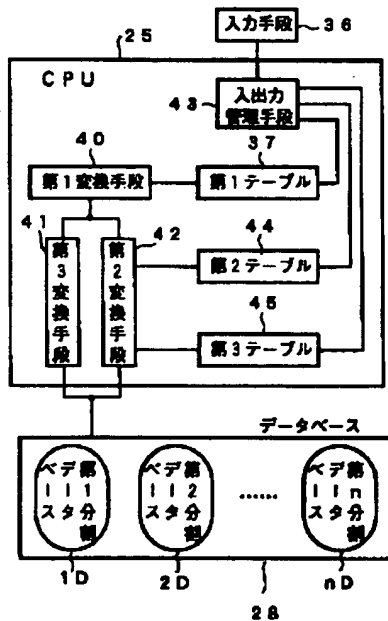
45 第3テーブル(列名定義テーブル)

1D~nD 第1~第n分割データベース

20

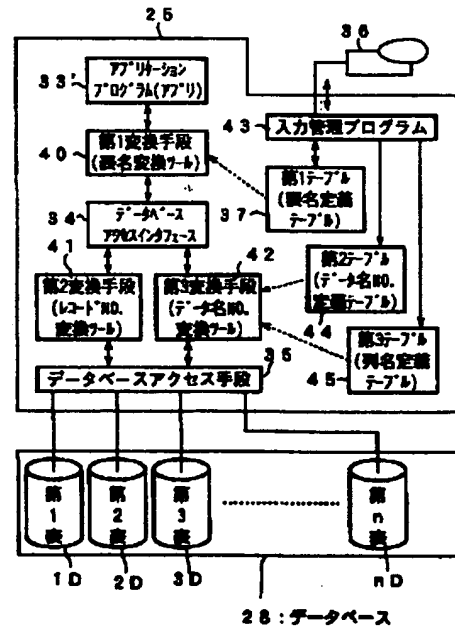
【図1】

本発明の原理図



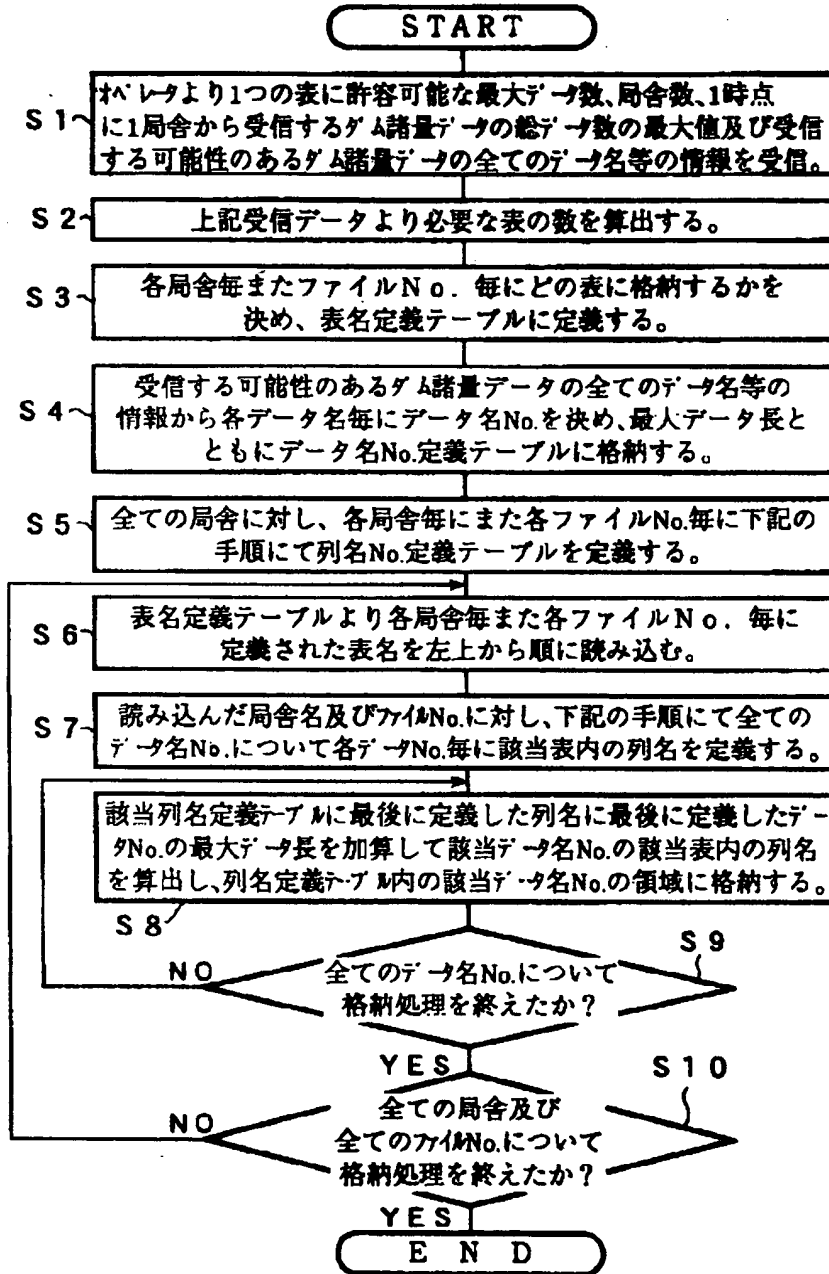
【図2】

本発明の一実施形態のCPUによるデータベースアクセス方式を適用したダム諸量システムにおける中央局のCPU機能ブロック構成図



【図3】

# 入出力管理プログラムがシステム立ち上げ時に行う 処理動作説明フローチャート



【図7】

図24に示す局舎FファイルNo. が月情報の場合の列名定義テーブル例図

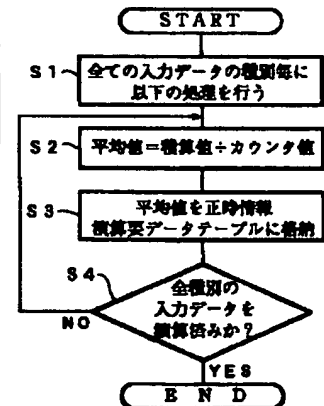
45

データ名No.	列名
1	3200
2	3380
3	3555
4	3725
5	3895
6	4065
...	...
n	XXXX

(局舎FファイルNo. が月情報)

【図17】

局舎内のCPUによる  
正時情報演算処理説明フローチャート



【図4】

データ名No. 定義テーブル例図

44

データ名	データ名No.	最大データ長
水位	1	180
貯水量	2	175
放流量	3	170
流入量	4	170
ゲート開度	5	179
調整流量	6	165
...	...	...
XXXXXX	n	XXX

【図5】

図24に示す局舎AファイルNo. が 正時情報の場合の列名定義テーブル例図

図24に示す局舎CファイルNo. が 正時情報の場合の列名定義テーブル例図

45

データ名No.	列名
1	0
2	180
3	355
4	525
5	695
6	865
...	...
n	XXXX

(局舎AファイルNo. が正時情報)

45

データ名No.	列名
1	0
2	180
3	355
4	525
5	695
6	865
...	...
n	XXXX

(局舎CファイルNo. が正時情報)

【図9】

データ名No. 定義テーブルの他例図

44

データ名	データ名No.	ファイルNo.	最大データ長
水位	1	正時情報	145
貯水量	2	正時情報	140
放流量	3	正時情報	135
流入量	4	正時情報	135
ゲート開度	5	正時情報	135
調整流量	6	正時情報	130
...	...	...	...
XXXXXX	n	正時情報	XXX
水位	1	日情報	160
貯水量	2	日情報	155
放流量	3	日情報	160
流入量	4	日情報	150
ゲート開度	5	日情報	150
調整流量	6	日情報	145
...	...	...	...
XXXXXX	n	日情報	XXX
水位	1	月情報	170
貯水量	2	月情報	165
放流量	3	月情報	160
流入量	4	月情報	160
ゲート開度	5	月情報	160
調整流量	6	月情報	155
...	...	...	...
XXXXXX	n	月情報	XXX
水位	1	年情報	180
貯水量	2	年情報	175
放流量	3	年情報	170
流入量	4	年情報	170
ゲート開度	5	年情報	170
調整流量	6	年情報	165
...	...	...	...
XXXXXX	n	年情報	XXX

【図10】

図24に示す局舎AファイルNo. が 正時情報の場合の列名定義テーブルの他例図

図24に示す局舎BファイルNo. が 正時情報の場合の列名定義テーブル例図

45

データ名No.	列名
1	0
2	145
3	285
4	420
5	555
6	690
...	...
n	XXXX

(局舎AファイルNo. が正時情報)

45

データ名No.	列名
1	2600
2	2745
3	2885
4	3020
5	3155
6	3290
...	...
n	XXXX

(局舎BファイルNo. が正時情報)